18.783 Elliptic Curves **Fall 2025**

Problem Set #1

These problems are related to material covered in Lectures 1–3. Some require the use of Sage; you will need to either create a (free) CoCalc account, or you can download and install a copy of Sage to run on your own computer (warning, it is big and takes time to setup, using CoCalc avoids this but won't work offline). Sage is based on the python programming language; you will find examples of Sage usage in the problems below, and there is a wealth of information to be found on the Sage website, including tutorials.

Instructions: Solve any combination of problems 1-4 that sums to 96 points, then complete the survey problem 6 (worth 4 points), whose results will help shape future problem sets and lectures. You can use the latex source for this problem set as a template for writing up your solutions. CoCalc includes an online latex editor, but feel free to use the latex environment of your choice (I personally use Overleaf and Texmaker, but there are many other options). Be sure to put your name on your solution (you can replace the due date in the header with your name). Your solutions should be written up in latex (please do not submit handwritten solutions) and submitted as a pdf-file .

Collaboration is permitted/encouraged, but you must identify your collaborators (including any LLMs you discussed the problem set with) or your group, as well any references you consulted that are not listed in the syllabus or lecture notes. Include this information after the **Collaborators/Sources** prompt at the end of the problem set (if there are none, you should enter "none", do not leave it blank). Each student is expected to write their own solutions; it is fine to discuss problems with others, but your writing must be your own.

Note: In this problem set k always denotes a field whose characteristic is not 2 or 3.

Problem 1. Montgomery arithmetic (32 points)

Let p be an odd prime. For any integers a, b with $b > 0$, let $a \operatorname{rem} b$ denote the unique integer in $[0, b-1]$ that is congruent to a modulo b, and let $a \operatorname{div} b$ denote the unique integer for which $a = (a \operatorname{div} b)b + a \operatorname{rem} b$; in other words, $a \operatorname{div} b$ and $a \operatorname{rem} b$ are the quotient and remainder in the Euclidean division of a by b.

For this problem we will assume we have a machine that performs integer arithmetic modulo 2^w for some fixed word-size w we shall assume is a power of 2 (typically $w = 64$ or $w = 128$) and that $p < 2^{w-1}$. For cryptographic applications one will have $p > 2^{w-1}$, but the algorithms we consider here can be readily extended to multi-word operands.

For integers $a, b \in [0, 2^w - 1]$ let $\mathbf{M}$ denote the cost of computing the pair of integers $(ab \operatorname{div} 2^w, ab \operatorname{rem} 2^w)$, and for $a, b \in [0, 2^{2w} - 1]$, let $\mathbf{A}$ denote the cost of computing $(a+b) \operatorname{rem} 2^w$, $(a-b) \operatorname{rem} 2^w$, or $a \operatorname{rem} 2^n$ for any $0 \le n \le w$. We assign a cost of $\mathbf{A} + \mathbf{M}$ to compute $(ab + c) \operatorname{div} 2^w$ with $a, b \in [0, 2^{w-1} - 1]$ and $c \in [0, 2^{2w-1} - 1]$.

For the purposes of this problem we will ignore the cost of operations that do not involve integer arithmetic; you can assume that setting an integer to 0 or 1, comparing integers, and testing a boolean condition all have zero cost.

In practice the cost of $\mathbf{A}$ is typically one clock cycle, the cost of $\mathbf{M}$ is typically 2-3 clock cycles; you can lookup the exact count for your favorite CPU here. We will assume that our computer does not directly support Euclidean division except by powers of 2. Modern computers do provide instructions for Euclidean division, but their cost is quite

high, 10-20 clock cycles or more for 64-bit operands. The goal of this problem is to show that we can perform ring operations in $\mathbb{Z}/p\mathbb{Z}$ without Euclidean division by p.

(a) Give an algorithm to compute $-1/p \operatorname{rem} 2^w$ with cost $2(\lg w)\mathbf{M} + (\lg w + 1)\mathbf{A}$ and an algorithm to compute $2^e \operatorname{rem} p$ with cost $2e\mathbf{A}$.

Define $q := -1/p \operatorname{rem} 2^w, R := 2^w \operatorname{rem} p, R^2 := 2^{2w} \operatorname{rem} p$. Henceforth we shall assume these quantities have been precomputed via **(a)**. The cost of this precomputation is not worth doing if you only want to perform one or two operations in $\mathbb{Z}/p\mathbb{Z}$, in which case you may as well use the Euclidean division modulo p provided by your CPU, but it is negligible for algorithms that perform many operations in $\mathbb{Z}/p\mathbb{Z}$.

(b) Let R^{-1} be the unique integer in $[0, p-1]$ for which $RR^{-1} \equiv 1 \bmod p$, and define

$$\operatorname{redc}(x) := xR^{-1} \operatorname{rem} p,$$

so that $\operatorname{redc}(x)R \equiv x \bmod p$. Show that given q, for any $x \in [0, p2^w - 1]$ we can compute $\operatorname{redc}(x)$ as

$$\operatorname{redc}(x) = (((((x \operatorname{rem} 2^w)q) \operatorname{rem} 2^w)p + x) \operatorname{div} 2^w) \operatorname{rem} p,$$

for a cost of $2\mathbf{M} + 3\mathbf{A}$.

Note that in your solution to **2b** you cannot use an operation of the form $a \operatorname{rem} p$, you are only allowed to use operations supported by our machine, and you need to prove that your solution works (you will need to use the assumption $x < p2^w$, for example).

I recommend implementing your "algorithm" in Sage (it will be just a few lines) to make sure you understand what is going on. The python syntax for $a \operatorname{rem} b$ is "`a%b`", and the syntax for $a \operatorname{div} b$ is "`a//b`" — note the double-slash, which tells python to do a Euclidean division, if you just type "`a/b`" you will get the rational number a/b.

Now let S be the set of integers in $[0, p-1]$ representing the ring $\mathbb{Z}/p\mathbb{Z}$ with addition and multiplication defined by $(a+b) \operatorname{rem} p$ and $ab \operatorname{rem} p$, respectively; for $a \in \mathbb{Z}$ the integer $a \operatorname{rem} p$ is the *standard representation* of a modulo p. Now let us define $[a] := a2^w \operatorname{rem} p$ as the *Montgomery representation*[1] of a modulo p.

(c) Show that the map $a \mapsto [a]$ is injective with inverse $[a] \mapsto \operatorname{redc}([a])$. This allows us to view the set $T := \{[a] : a \in S\} \subseteq [0, 2^w - 1]$ as an alternative representation of the ring $\mathbb{Z}/p\mathbb{Z}$ with addition and multiplication defined by $[a] \oplus [b] := [(a+b) \operatorname{rem} p]$ and $[a] \otimes [b] := [ab \operatorname{rem} p]$ (here $\oplus$ and $\otimes$ denote addition and multiplication in T).

Show that $[0] = 0$, $[1] = R$, and explain how to compute in the ring T for a cost of $2\mathbf{A}$ per addition and $3\mathbf{M} + 3\mathbf{A}$ per multiplication, assuming q has been precomputed (in practice the cost of multiplication is closer to $3\mathbf{M} + \mathbf{A}$; some bit-shifting operations we are charging for actually have zero cost).

(d) The map $[a] \mapsto \operatorname{redc}([a]) = a$ allows us to recover a from its Montgomery representation $[a]$, but this still leaves the question of how to compute $[a] := a2^w \operatorname{rem} p$. Give an algorithm that computes $[a]$ given a and R^2 for a cost of $3\mathbf{M} + 3\mathbf{A}$.

[1] Named after Peter L. Montgomery, also known for the *Montgomery ladder* we will see later.

Problem 2. Twists of elliptic curves (32 points)

Let E/k be an elliptic curve in short Weierstrass form

$$E: \qquad y^2 = x^3 + Ax + B.$$

The *quadratic twist* of E by $c \in k^\times$ is the elliptic curve over k defined by the equation

$$E_c: \qquad cy^2 = x^3 + Ax + B.$$

We will formally define morphisms of elliptic curves in Lecture 4 as morphisms of curves (these are defined by rational maps) that send the distinguished point of the domain to the distinguished point of the codomain. This includes linear changes of variable defined by $x \mapsto rx + s$ and $y \mapsto ty + ux + v$ with $r, t \in k^\times$ and $s, u, v \in k$, and for elliptic curves defined by an affine equation of the form $g(y) = f(x)$ with $\deg g = 2$, $\deg f = 3$, and the distinguished point at infinity, these are the only possibilities (other than multiplying both sides of the equation by an element of $k^\times$; recall that for a plane curve $f(x, y, z) = 0$ the polynomial f is determined only up to a scalar). It is not hard to show that isomorphisms $E_c \xrightarrow{\sim} E_{c'}$ require $s = u = v = 0$, which you may assume.

(a) Using a linear change of variables, show that E_c is isomorphic to an elliptic curve in standard Weierstrass form $y^2 = x^3 + A'x + B'$, and express A' and B' in terms of A and B and c. Verify that E_c is not singular.

(b) For any group G and positive integer n, we use $G[n]$ to denote the n-torsion subgroup of G, consisting of all elements whose order divides n. Prove that $E(k)[2] = E_c(k)[2]$.

(c) Prove that if c is a square in $k^\times$, then E and E_c are isomorphic over k. Conclude that E and E_c are always isomorphic over $k(\sqrt{c})$, whether c is a square in $k^\times$ or not (in general, curves defined over k are *twists* if they are isomorphic over some extension of k).

(d) Show that when $B = 0$, replacing A by $A' := cA$ for some nonsquare c yields an elliptic curve E' that is not necessarily a quadratic twist of E but is a *quartic twist* of E, which becomes isomorphic to E over the extension $k(c^{1/4})$. Similarly show how to construct *cubic twists* and *sextic twists* of E when $A = 0$.

(e) Now let $k = \mathbb{F}_q$ be a finite field of odd characteristic, and let t be the unique integer for which

$$\#E(\mathbb{F}_q) = q + 1 - t,$$

where $\#E(\mathbb{F}_q)$ is the cardinality of the group of $\mathbb{F}_q$-rational points of E. Prove that

$$\#E_c(\mathbb{F}_q) = q + 1 - \chi(c)t,$$

where $\chi\colon \mathbb{F}_q^\times \to \{\pm 1\}$ is the quadratic character of $\mathbb{F}_q^\times$ (so $\chi(c) = 1$ iff c is a square).

(f) Continuing with $k = \mathbb{F}_q$, show that if $t \neq 0$ then E_c and $E_{c'}$ are isomorphic if and only if $\chi(c) = \chi(c')$.

Problem 3. Four torsion subgroups (32 points)

Let E/k be an elliptic curve in short Weierstrass form

$$E: \qquad y^2 = f(x) = x^3 + Ax + B,$$

and let $f'(x) := 3x^2 + A$ denote the formal derivative of $f(x)$. Let $E[n] := E(\bar{k})[n]$ denote the n-torsion subgroup of $E(\bar{k})$. The goal of this problem is to gain a better understanding of the 2-torsion and 4-torsion subgroups of E.

You may want to solve (or at least read) Problem 2 before attempting this problem, particularly **(2b)**.

(a) Prove that $P \in E(\bar{k})$ has order 2 if and only if $P = (x_0, 0)$ with $f(x_0) = 0$. Conclude that $E[2] \simeq \mathbb{Z}/2\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z}$ and $E[2^r] \simeq \mathbb{Z}/2^r\mathbb{Z} \oplus \mathbb{Z}/2^r\mathbb{Z}$ for all $r \geq 1$.

(b) Let $Q = (x_0, 0) \in E(\bar{k})$ and let $P = (u, v) \in E(\bar{k})$. Prove that $2P = Q$ if and only if we have $f'(x_0) = (u - x_0)^2$.

Now let $k = \mathbb{F}_q$ be a finite field of odd characteristic and let $\chi\colon \mathbb{F}_q^\times \to \{\pm 1\}$ be its quadratic character.

(c) Prove that $E(\mathbb{F}_q)$ contains a point of order 4 only if $\chi(f'(x_0)) = 1$ for some rational root $x_0 \in \mathbb{F}_q$ of $f(x)$. Show that this necessary condition is not always sufficient.

(d) Suppose that $f(x)$ has three rational roots $x_1, x_2, x_3 \in \mathbb{F}_q$. Prove that

$$\chi(-1)\chi(f'(x_1))\chi(f'(x_2))\chi(f'(x_3)) = 1.$$

Conclude that if $q \equiv 3 \bmod 4$ then $E(\mathbb{F}_q)[4] \neq E[4]$.

Let $c \in \mathbb{F}_q^\times$ be a nonsquare and let E_c denote the quadratic twist of E, as in Problem 3. If $f(x)$ has no rational roots then $E(\mathbb{F}_q)[4] = E_c(\mathbb{F}_q)[4] = \{0\}$ (by **3b** and **4a**).

(e) Determine up to isomorphism the unordered pairs $(E(\mathbb{F}_q)[4], E_c(\mathbb{F}_q)[4])$ that can arise when $f(x)$ has exactly one rational root, where q and $f(x)$ are allowed to vary subject to this constraint.

(f) Determine up to isomorphism the unordered pairs $(E(\mathbb{F}_q)[4], E_c(\mathbb{F}_q)[4])$ that can arise when $f(x)$ has three rational roots and $q \equiv 3 \bmod 4$, and then do the same for $q \equiv 1 \bmod 4$, where q and f are allowed to vary subject to these constraints.

Problem 4. Sato-Tate for CM elliptic curves (32 points)

Recall from Lecture 1 that the elliptic curve $E/\mathbb{Q}$ defined by $y^2 = x^3 + Ax + B$ has *good reduction* at a prime p whenever p does not divide $\Delta(E) := -16(4A^3 + 27B^2)$. For each prime p of good reduction, let

$$a_p = p + 1 - \#E_p(\mathbb{F}_p) \qquad \text{and} \qquad x_p = a_p/\sqrt{p},$$

where E_p denotes the reduction of E modulo p.

To create an elliptic curve defined by a short Weierstrass equation in Sage, you can type `E=EllipticCurve([A,B])`. To check whether the elliptic curve E has good reduction at p, use `E.has_good_reduction(p)`, and to compute a_p, use `E.ap(p)`.

In this problem you will investigate the distribution of x_p for some elliptic curves over $\mathbb{Q}$ to which the Sato-Tate conjecture does not apply. These are elliptic curves with *complex multiplication* (CM for short), a term we will define later in the course. In Sage you can check for CM using `E.has_cm()`.

(a) Let $E/\mathbb{Q}$ be the curve defined by $y^2 = x^3 + 1$. Compute a list of a_p values for the primes $p \le 200$ where E has good reduction (all but 2 and 3). The following block of Sage code does this.

```
E=EllipticCurve([0,1])
for p in primes(0,200):
    if E.has_good_reduction(p):
        print("%d:%d" % (p, E.ap(p)))
```

You will notice that many of the a_p values are zero. Give a conjectural criterion for the primes p for which $a_p = 0$. Verify your conjecture for all primes $p \le 2^{10}$ where E has good reduction.

(b) Given a bound B, the nth *moment statistic* M_n of x_p is defined as the average value of x_p^n over primes $p \le B$ where E has good reduction. In Lecture 1 we saw that for an elliptic curve over $\mathbb{Q}$ without complex multiplication, the sequence of moment statistics $M_0, M_1, M_2, \ldots$ appear to converge to the integer sequence

$$1, 0, 1, 0, 2, 0, 5, 0, 14, 0, 42, \ldots,$$

whose odd terms are 0 and whose even terms are the Catalan numbers. Your goal is to determine an analogous sequence for elliptic curves over $\mathbb{Q}$ with complex multiplication.

To do this efficiently, use the `E.aplist()` method in Sage. The following block of code computes the moment statistics $M_0, \ldots, M_{10}$ of x_p using the bound $B = 2^k$.

```
k=12
E=EllipticCurve([0,1])
A=E.aplist(2^k)
P=prime_range(0,2^k)
X=[A[i]/sqrt(RR(P[i])) for i in range(0,len(A))]
M=[sum([a^n for a in X])/len(X) for n in [0..10]]
print(M)
```

(note that use of `RR(P[i])` to coerce the prime `P[i]` to a real number before taking its square root — without this Sage will use a symbolic representation of the square root as an algebraic number, which is not what we want). With this approach we are also including a few a_p values at bad primes (which will yield $x_p \approx 0$), but this is harmless as long as we make $B = 2^k$ large enough.

By computing moment statistics using bounds $B = 2^k$ with $k = 12, 16, 20, 24$, determine the integers to which the first ten moment statistics appear to converge, and come up with a conjectural formula for the nth moment (if you get stuck on this, look at **(e)** and **(f)** below). Then test your conjecture by computing the 12th and 14th moment statistics and comparing the results.

(c) Repeat the analysis in parts **(a)** and **(b)** for the following elliptic curves over $\mathbb{Q}$:

$$y^2 = x^3 - 595x + 5586,$$
$$y^2 = x^3 - 608x + 5776,$$
$$y^2 = x^3 - 9504x + 365904.$$

You will probably need to look at more a_p values than just up to $p \leq 200$ in order to formulate a criterion for the a_p that are zero. Do the x_p moment statistics for these elliptic curves appear to converge to the same sequence you conjectured in part **(b)**?

(d) Pick one of the three curves from part **(c)** and take its quadratic twist by the last four digits of your student ID. Does this change the sequence of a_p values? Does it change the moment statistics of x_p?

(e) Recall that the special orthogonal group SO(2) consists of all matrices of the form $R_\theta = \left(\begin{smallmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{smallmatrix}\right)$. To generate a random matrix in SO(2), one simply picks θ uniformly at random from the interval $[0, 2\pi)$; this is the *Haar measure* on SO(2), the unique probability measure that is invariant under the group action. Derive a formula for the nth moment of the trace of a random matrix in SO(2) by integrating the nth power of the trace of R_θ over all $\theta \in [0, 2\pi)$. Be sure to normalize by $1/(2\pi)$ so that $M_0 = 1$.

(f) The normalizer $N(\mathrm{SO}(2))$ of SO(2) in the special unitary group SU(2) consists of all matrices of the form R_θ and JR_θ, where $J = \left(\begin{smallmatrix} i & 0 \\ 0 & -i \end{smallmatrix}\right)$. Derive a formula for the nth moment of the trace of a random matrix in $N(\mathrm{SO}(2))$ (under the Haar measure on $N(\mathrm{SO}(2))$ one picks $\theta \in [0, 2\pi)$ uniformly at random and then takes R_θ or JR_θ with equal probability). Compare the results to the formula you conjectured in part **(b)**.

Problem 6. Survey (4 points)

Complete the following survey by rating each of the problems you solved on a scale of 1 to 10 according to how interesting you found the problem (1 = "mind numbing," 10 = "mind blowing"), and how difficult you found the problem (1 = "trivial," 10 = "brutal"). Also estimate the amount of time you spent on each problem to the nearest half hour.

	Interest	Difficulty	Time
Problem 1			
Problem 2			
Problem 3			
Problem 4			

Feel free to record any additional comments you have on the problem sets or lectures; in particular, how you think they could be improved (which they surely can!).

Collaborators/sources:

18.783 / 18.7831 Elliptic Curves
Fall 2025